WEST Search History

DATE: Monday, November 17, 2003

Set Name side by side	Query	Hit Count	Set Name result set
DB=JPAB,EPAB,DWPI,TDBD; PLUR=NO; OP=ADJ			
L26	l24 not L25	76	L26
L25	L24 not l22	338	L25
L24	instruction$1 with L23	414	L24
L23	(load$3 or stor$3) with L21	41855	L23
L22	L21 not (packet$1 or package$1 or packard)	231362	L22
L21	pack$3 or unpack$3 or un-pack$3	527166	L21
DB=USPT,PGPB; PLUR=NO; OP=ADJ			
L20	l14 not L19	311	L20
L19	l14 not l17	2118	L19
L18	l14 not l17L17	2429	L18
L17	l15 not package$1	246928	L17
L16	l14 and L15	753	L16
L15	l12 not (packet$1)	453221	L15
L14	instruction$1 with L13	2429	L14
L13	(load$3 or stor$3) with L12	72868	L13
L12	pack$3 or unpack$3 or un-pack$3	533925	L12
L11	load lower byte	4	L11
L10	load packed byte	0	L10
L9	load unpacked data	0	L9
DB=JPAB,EPAB,DWPI,TDBD; PLUR=NO; OP=ADJ			
L8	load packed word	0	L8
L7	load packed byte	0	L7
L6	load lower byte	0	L6
L5	load upper byte	0	L5
L4	load packed data	0	L4
L3	load unpacked data	0	L3
L2	unpack instruction$1	8	L2
DB=USPT,PGPB; PLUR=NO; OP=ADJ			
L1	unpack instruction$1	67	L1

END OF SEARCH HISTORY

WEST

Generate Collection | Print

L20: Entry 106 of 311 File: USPT Oct 23, 2001

US-PAT-NO: 6307553
DOCUMENT-IDENTIFIER: US 6307553 B1
**** See image for Certificate of Correction ****

TITLE: System and method for performing a MOVHPS-MOVLPS instruction

DATE-ISSUED: October 23, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Abdallah; Mohammad	Folsom	CA	94630	

APPL-NO: 09/ 053001 [PALM]
DATE FILED: March 31, 1998

INT-CL: [07] G06 T 1/00

US-CL-ISSUED: 345/419; 712/220, 712/221, 712/222
US-CL-CURRENT: 345/419; 712/220, 712/221, 712/222

FIELD-OF-SEARCH: 345/419, 712/220-222

PRIOR-ART-DISCLOSED:

OTHER PUBLICATIONS

Sun Microsystems Inc. Technology White Paper: The UltraSPARC Architecture, Nov. 1995.*
Visual Instruction Set User's Guide, Mar. 1997.*
Sun Microsystems, "Visual Instruction Set User's Guide", Mar. 1997.
Tri Media, "TM 1000 Preliminary Data Book", Phillips Electronics North America Corporation, 1997.
Wang, Mangaser, Srinivasan, "A Processor Architecture for 3D Graphics Calculations", Computer Motion Inc., Goleta, CA.
Abott, Massalin, Peterson, Karzes, Yamano, Kellogg, "Broadband Algorithms with the MicroUnity Mediaprocessor", Proceedings of Compcon, IEEE, 1996, pp. 349-354.
Advanced Micro Devices, Inc., "AMD-3D Technology Manual", Feb., 1998.
Hansen, Craig, "Architecture of a Broadband Mediaprocessor", Proceedings of Compcon, IEEE, 1996, pp. 334-340.
Hayes, Loyola, Abott, Massalin, "MicroUnity Software Development Environment", Proceedings of Compcon, IEEE, 1996, pp. 341-348.
Levinthal, Hanrahan, Paquette, Lawson, "Parallel Computers for Graphics Applications", Proceedings of: ASPLOS II, IEEE, 1987, pp. 193-198.
Levinthal and Porter, "Chap-A SIMD Graphics Processor", Computer Graphics Project,, ACM, vol. 18, No. 3, Jul. 1984, pp. 77-81.

ART-UNIT: 272

PRIMARY-EXAMINER: Powell; Mark R.

ASSISTANT-EXAMINER: Sealey; Lance W.

ABSTRACT:

An apparatus and method for performing a MOVHPS-MOVLPS operation on packed data using computer-implemented steps is described. In one embodiment, a first packed data operand having a pair of data elements is accessed. A second packed data operand having two pairs of data elements is then accessed. One of the two pairs of data elements in the second packed data operand is replaced with the pair of data elements in the first packed data operand.

10 Claims, 8 Drawing figures

WEST

Generate Collection | Print

L20: Entry 106 of 311 File: USPT Oct 23, 2001

DOCUMENT-IDENTIFIER: US 6307553 B1
**** See image for Certificate of Correction ****
TITLE: System and method for performing a MOVHPS-MOVLPS instruction

Detailed Description Text (3):
According to one aspect of the invention, a method and apparatus are described for moving data elements in a packed data operand (a MOVHPS-MOVLPS operation). The MOVHPS-MOVLPS operations allow, for example, the partial update of a 128-bit packed register from memory, and the partial store of the 128-bit register into 64-bit memory. This allows the upper half or lower half of the register/memory to be bypassed to destination without modification. This has the benefit of 1) potentially achieving the same performance when updating a 128-bit register or memory as a packed instruction implementation, and 2) providing the flexibility of loading into the packed register from different 64-bit memory locations all storing from two different packed memory locations. The two halves of the 128-bit register may be assembled with the same performance as the packed instruction which loads or stores an entire 128-bit register to/from a unified 128-bit memory location. Being able to access a 64-bit quantity, rather than a full 128-bit quantity, is also useful when reorganizing data formats.

Detailed Description Text (10):
The decode unit 140 is shown including packed data instruction set 145 for performing operations on packed data. In one embodiment, the packed data instruction set 145 includes the following instructions: a move instruction(s) 150, a shuffle instruction(s) 155, an add instruction(s) (such as ADDPS) 160, and a multiply instruction(s) 165. The MOVAPS, SHUFPS and ADDPS instructions are applicable to packed floating point data, in which the results of an operation between two sets of numbers having a predetermined number of bits, are stored in a register having the same predetermined number of bits, i.e., the size or configuration of the operand is the same as that of the result register. The operation of each of these instructions is further described herein. While one embodiment is described in which the packed data instructions operate on floating point data, alternative embodiments could alternatively or additionally have similar instructions that operate on integer data.

WEST

Generate Collection | Print

L20: Entry 121 of 311 File: USPT Jul 10, 2001

US-PAT-NO: 6260137
DOCUMENT-IDENTIFIER: US 6260137 B1

TITLE: Data processing unit with digital signal processing capabilities

DATE-ISSUED: July 10, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Fleck; Rod G.	Mountain View	CA		
Martin; Daniel	Mountain View	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Siemens Aktiengesellschaft	Munich			DE	03

APPL-NO: 08/ 928764 [PALM]
DATE FILED: September 12, 1997

INT-CL: [07] G06 F 9/315

US-CL-ISSUED: 712/225; 712/224
US-CL-CURRENT: 712/225; 712/224

FIELD-OF-SEARCH: 395/393, 395/588, 395/598, 395/800.13, 395/800.22, 395/800.23, 395/800.24, 395/800.35, 395/800.36, 395/800.41, 395/380, 395/306, 395/316, 712/217, 712/241, 712/248, 712/1-43, 712/2T, 712/224, 712/225, 710/9, 711/1-6, 711/1T

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
4992934	February 1991	Portanova et al.	712/209
5269007	December 1993	Hanawa	712/218
5367705	November 1994	Sites	710/21
5574928	November 1996	White	712/23
5721892	February 1998	Peleg	712/221
5734874	March 1998	Van Hook	345/513
5752271	May 1998	Yung	712/23
5768609	June 1998	Gove et al.	712/11
5778241	July 1998	Bindloss et al.	712/20
5812147	September 1998	Van Hook	345/511
5852726	December 1998	Lin et al.	712/200
5864713	January 1999	Terry	395/872
5896543	April 1999	Garde	712/35
5913054	June 1999	Mallick et al.	711/200
5918252	June 1999	Chen et al.	711/217
5983256	November 1999	Peleg et al.	708/523

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
0 473 805 A1	September 1990	EP	
0 679 991 A1	November 1995	EP	
WO 96/17291	June 1996	WO	

OTHER PUBLICATIONS

Sun Microsystems, "Ultrasparc-I,-II User's manual", pp. 190-234, [retrieved on Jul. 7, 1999]. Retieved from the Internet:<URL: http://www.sun.com/microelectronics/UltraSPARC-II/; $sessionid$2EHV1ZQAAFGW5AMUVFZE5YQ.*
Sun Microsystems, "Ultrasparc-IIi User's manual", pp.127-133, Oct. 1997.

ART-UNIT: 282

PRIMARY-EXAMINER: Niebling; John F.

ASSISTANT-EXAMINER: Whitmore; Stacy

ABSTRACT:

The present invention relates to a data processing unit comprising a register file, a register load and store buffer connected to the register file, a single memory, and a bus having at least first and second word lines to form a double word wide bus coupling the register load and store buffer with said single memory. The register file at least two sets of registers whereby the first set of registers can be coupled with one of the word lines and the second set of registers can be coupled with the respective other word lines, a load and store control unit for transferring data from or to the memory.

24 Claims, 9 Drawing figures

WEST

Generate Collection | Print

L20: Entry 121 of 311 File: USPT Jul 10, 2001

DOCUMENT-IDENTIFIER: US 6260137 B1
TITLE: Data processing unit with digital signal processing capabilities

Detailed Description Text (8):
A second type of instruction which can be executed according to the present invention is a so called "load two half-words (packed)"-instruction. With this instruction one word from either data lines 1a or 1d is loaded and split into half-words by units 8 or 9 placed in the respective lower halves of a word. Optionally units 12 and 13 can either sign-extend or zero-extend the respective half-words to words. In other words, in this embodiment, the 16 bit half-words are extended to 32 bits. Unit 8 or unit 9 splits the word received from lines 1a or 1d into two half-words and distributes them through units 12 and 13 to the lower halves of the respective even and odd registers. In units 12 and 13 these half-words can be extended to words either by filling the upper halves with zeros or by sign extending the upper halves. If the sign of a half-word is negative the upper halves of the respective register is filled up with "1" otherwise with "0". If units 12 and 13 are deactivated the half-words are stored into the lower halves of the respective even and odd registers without changing their upper halves. In a simplified version the least significant memory half-word is always stored into an even register and the most significant half-word is stored into an odd register adjacent to the even register.

Detailed Description Text (10):
A fourth type of instruction which can be executed according to the present invention is a so called "store two half-words (packed)"-instruction. With this instruction the lower half-words of an even and an odd register are fed to either concatenating unit 11 or 14. The two half-words are combined to one word and the stored in the memory unit 1 through multiplexer 7 or 10 and either data input lines 1b or 1c.

Detailed Description Text (19):
This so called packed arithmetic or logical instructions partition, in this embodiment, a 32 bit word into several identical objects, which can then be fetched, stored, and operated on in parallel. These instructions, in particular, allow the full exploitation of the 32 bit word of the data processing unit according to the present invention in DSP applications.

Detailed Description Text (21):
The loading and storing of packed values into data or address registers is supported by the respective load and store instructions described above. The packed objects can then be manipulated in parallel by a set of special packed arithmetic instructions that perform such arithmetic operations as addition, subtraction, multiplication, division, etc. For example a multiply instruction performs two, 16 bit multiplication's in parallel as shown in FIG. 5.

WEST

Generate Collection | Print

L20: Entry 140 of 311 File: USPT Jan 9, 2001

US-PAT-NO: 6173366
DOCUMENT-IDENTIFIER: US 6173366 B1
**** See image for Certificate of Correction ****

TITLE: Load and store instructions which perform unpacking and packing of data bits in separate vector and integer cache storage

DATE-ISSUED: January 9, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Thayer; John S.	Houston	TX		
Favor; John G.	Scotts Valley	CA		
Weber; Frederick D.	San Jose	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Compaq Computer Corp.	Houston	TX			02
Advanced Micro Devices, Inc.	Sunnyvale	CA			02

APPL-NO: 08/ 759044 [PALM]
DATE FILED: December 2, 1996

INT-CL: [07] G06 F 12/04

US-CL-ISSUED: 711/129; 711/125, 712/4, 710/66, 710/68, 710/130
US-CL-CURRENT: 711/129; 710/66, 710/68, 711/125, 712/4

FIELD-OF-SEARCH: 711/118, 711/123, 711/125, 711/131, 711/220, 711/211, 711/217, 711/171, 711/173, 711/129, 345/202, 345/520, 345/521, 345/193, 345/198, 712/3, 712/4, 710/50, 710/66, 710/68, 710/130

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
3541516	November 1970	Senzig	714/6
3701977	October 1972	Mendelson et al.	711/126
4783736	November 1988	Ziegler et al.	711/130
4884197	November 1989	Sachs et al.	711/123
4891754	January 1990	Boreland	
5025407	June 1991	Gulley et al.	
5193167	March 1993	Sites et al.	
5307300	April 1994	Komoto et al.	
5335330	August 1994	Inoue	712/241
5437043	July 1995	Fujii et al.	
5481713	January 1996	Wetmore et al.	395/705
5513366	April 1996	Agarwal et al.	712/22
5627981	May 1997	Adler et al.	
5640588	June 1997	Vegesna et al.	
5669013	September 1997	Watanabe et al.	
5801975	September 1998	Thayer et al.	
5845083	December 1998	Hamadani et al.	709/231
5893145	April 1999	Thayer et al.	
5909572	June 1999	Thayer et al.	

OTHER PUBLICATIONS

Mahlke et al., "A Comparison of Full and Partial Prdicated Execution Support for ILP Processor" by 1995, IEEE Publication, pp. 138-149.
Kohn, L, et al., "The Visual Instruction Set (VIS) in UltraSPARC," SPARC Technology Business--Sun Microsystems, Inc., 1996 IEEE, pp. 462-489.
Gwennap, Linley, "UltraSparc Adds Multimedia Instructions--Other New Instructions Handle Unaligned and Little-Endian Data," Microprocessor Report, Dec. 5, 1994, pp. 16-18.
Lee, Ruby B., "Realtime MPEG Video via Software Decompression on a PA-RISC Processor," Hewlett-Packard Company, 1995 IEEE, pp. 186-192.
Mattison, Phillip E., "Practical Digital Video With Programming Examples in C," Wiley Professional Computing, pp. 158-178.
Zhou, Chang-Guo, et al., "MPEG Video Decoding With the UltraSPARC Visual Instruction Set," Sun Microsystems, Inc., 1995 IEEE, pp. 470-474.

ART-UNIT: 272

PRIMARY-EXAMINER: Peikari; B. James

ATTY-AGENT-FIRM: Conley, Rose & Tayon, P.C. Kowert; Robert C. Daffer; Kevin L.

ABSTRACT:

A multimedia extension unit (MEU) is provided for performing various multimedia-type operations. The MEU can be coupled either through a coprocessor bus or a local CPU bus to a conventional processor. The MEU employs vector registers, a vector ALU, and an operand routing unit (ORU) to perform a maximum number of the multimedia operations within as few instruction cycles as possible. Complex algorithms are readily performed by arranging operands upon the vector ALU in accordance with the desired algorithm flowgraph. The ORU aligns the operands within partitioned slots or sub-slots of the vector registers using vector instructions unique to the MEU. At the output of the ORU, operand pairs from vector source or destination registers can be easily routed

and combined at the vector ALU. The vector instructions employ special load/store instructions in combination with numerous operational instructions to carry out concurrent multimedia operations on the aligned operands.

31 Claims, 19 Drawing figures

WEST

Generate Collection Print

L20: Entry 140 of 311 File: USPT Jan 9, 2001

DOCUMENT-IDENTIFIER: US 6173366 B1
**** See image for Certificate of Correction ****
TITLE: Load and store instructions which perform unpacking and packing of data bits in separate vector and integer cache storage

Brief Summary Text (41):
Arithmetic scaling which is lacking from many conventional operations is readily performed as part of the present load/store instructions. For example, packing and unpacking instructions found in many DSP instruction sets can be avoided. Thus, unpacking of an 8-bit word into a 20-bit slot occurs as part of a load instruction, whereas packing of a 20-bit operand to an 8-bit word occurs as part of the store instruction. Combining packing and unpacking operations into store and load helps eliminate unnecessary move operations which occur as part of stand-alone conventional pack and unpack instructions.

Detailed Description Text (109):
The interleave mapping for 10-bit partitions is completely transparent to the programmer as long as only 10-bit loads/stores and vector instructions are performed on a given set of data. Interleaved mapping of 20-bit partitions is also transparent to the programmer if only 20-bit operations are performed. However, if 10-bit and 20-bit operations are mixed, then care must be taken to understand the mapping so that the expected results are produced. The interleaving can be very useful, for example, if a 10-bit load from an octet-sized memory location automatically expands and interleaves the byte-wide memory data to the upper portion of 20-bit partitions. The 20-bit operation can be immediately performed on this data without the need for explicit format conversions. Subsequently, 10-bit stores to octets can automatically perform the inverse 20-bit to 10-bit packing function. Thus, the present store operation, namely vstb mem64, vsh performs packing of n+4 bits within a slot of a vector register to n/2 bits within an address of the memory unit. Given n=16, 20-bit-to-8-bit packing can occur as part of the store operation. Additional operations, such as move or shift operations need not occur to perform a packing function. Packing serves to store the most significant bits from a slot. Unpacking is an operation by which n/2 bits from a memory address are loaded into n+4 bit locations within a slot. If n=16, then a load operation such as vldb vdh, mem64 causes 8-bits within a memory address to be loaded into a 20-bit slot. Utilizing load and store functions in such a manner thereby avoids having to implement separate unpack and pack instructions, respectively, within the MEU instruction set. Accordingly, the same result can be achieved but with fewer instructions. For MPEG, 8-bit pixels are unpacked to 20-bit numbers for DCT or IDCT manipulations, then the results are repacked to 8-bit pixels. The internals of the DCT and IDCT operations require more than 8 bits of precision, to which packing and unpacking are particularly advantageous.

CLAIMS:

1. A computer, comprising:

an input/output device operably coupled to a microprocessor, wherein the microprocessor includes:

an instruction cache configured to store coded first and second sets of instructions obtained from the input/output device, wherein said first set of instructions comprises integer instructions for operating on integer operands and said second set

of instructions comprises vector instructions for operating on vector data;

a decode unit configured to decode said vector instructions; and

wherein said microprocessor is configured to perform a load of data having a first bit size from a first memory location having said first bit size to a register slot having a second bit size, wherein said first bit size is smaller than said second bit size, and wherein said microprocessor is configured to perform an unpacking operation during the load to fill said register slot, wherein said microprocessor is configured to load said data and perform said unpacking operation in response to said decode unit decoding a single vector load instruction.

6. The computer as recited in claim 1, wherein said unpacking operation occurs within the same instruction cycle as the vector load instruction.

8. A computer, comprising:

an input/output device operably coupled to a microprocessor, wherein the microprocessor comprises:

an instruction cache configured to store coded first and second sets of instructions obtained from the input/output device, wherein said first set of instructions comprises integer instructions for operating on integer operands and said second set of instructions comprises vector instructions for operating on vector data;

a decode unit configured to decode said vector instructions; and

wherein said microprocessor is configured to perform a store of data having a second bit size from a register slot having said second bit size to a first memory location having a first bit size, wherein said first bit size is smaller than said second bit size, and wherein said microprocessor is configured to perform a packing operation during the store on said data to fit said data into said first memory location, wherein said microprocessor is configured to store said data and perform said packing operation in response to said decode unit decoding a single vector store instruction.

12. The computer as recited in claim 8, wherein said packing operation occurs within the same instruction cycle as the vector store instruction.

13. A microprocessor, comprising:

an instruction cache configured to store coded first and second sets of instructions, wherein said first set of instructions comprises integer instructions for operating on integer operands and said second set of instructions comprises vector instructions for operating on vector data;

a decode unit configured to decode said vector instructions; and

wherein said microprocessor is configured to perform a load of data having a first bit size from a first memory location having said first bit size to a register slot having a second bit size, wherein said first bit size is smaller than said second bit size, and wherein said microprocessor is configured to perform an unpacking operation during the load to fill said register slot, wherein said microprocessor is configured to load said data and perform said unpacking operation in response to said decode unit decoding a single vector load instruction.

18. The microprocessor as recited in claim 13, wherein said unpacking operation occurs within the same instruction cycle as the vector load instruction.

24. A microprocessor, comprising:

an instruction cache configured to store coded first and second sets of instructions, wherein said first set of instructions comprises integer instructions for operating on integer operands and said second set of instructions comprises vector instructions for operating on vector data;

a decode unit configured to decode said vector instructions; and

wherein said microprocessor is configured to perform a store of data having a second bit size from a register slot having said second bit size to a first memory location having a first bit size, wherein said first bit size is smaller than said second bit size, and wherein said microprocessor is configured to perform a packing operation during the store on said data to fit said data into said first memory location, wherein said microprocessor is configured to store said data and perform said packing operation in response to said decode unit decoding a single vector store instruction.

28. The microprocessor as recited in claim 24, wherein said packing operation occurs within the same instruction cycle as the vector store instruction.

WEST

Generate Collection Print

L20: Entry 151 of 311 File: USPT Aug 8, 2000

US-PAT-NO: 6101592
DOCUMENT-IDENTIFIER: US 6101592 A

TITLE: Methods and apparatus for scalable instruction set architecture with dynamic compact instructions

DATE-ISSUED: August 8, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Pechanek; Gerald G.	Cary	NC		
Barry; Edwin F.	Cary	NC		
Revilla; Juan Guillermo	Cary	NC		
Larsen; Larry D.	Raleigh	NC		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Billions of Operations Per Second, Inc.	Chapel Hill	NC			02

APPL-NO: 09/ 215081 [PALM]
DATE FILED: December 18, 1998

PARENT-CASE:
CROSS-REFERENCE TO RELATED APPLICATION The present application claims the benefit of U.S. Provisional Application Serial No. 60/068,021 entitled "Methods and Apparatus for Scalable Instruction Set Architecture" and filed Dec. 18, 1997.

INT-CL: [07] G06 F 15/80

US-CL-ISSUED: 712/20; 712/22, 712/24, 712/209, 712/212
US-CL-CURRENT: 712/20; 712/209, 712/212, 712/22, 712/24

FIELD-OF-SEARCH: 712/20, 712/21, 712/22, 712/24, 712/209, 712/212

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	5057837	October 1991	Colwell et al.	341/55
☐	5555428	September 1996	Radigan et al.	712/24
☐	5649135	July 1997	Pechanek et al.	
☐	5761470	June 1998	Yoshida	712/210
☐	5787302	July 1998	Hampapuram et al.	712/24
☐	5819058	October 1998	Miller et al.	712/210
☐	5822606	October 1998	Morton	712/24
☐	5870576	February 1999	Faraboschi et al.	712/24
☐	5901301	May 1999	Matsuo et al.	712/212
☐	5926644	July 1999	Hays	712/24
☐	5930508	July 1999	Faraboschi et al.	712/24
☐	5983336	November 1999	Sakhin et al.	712/24

OTHER PUBLICATIONS

Andrew Tanenbaum , Structured Computer Organization, Prentice Hall, pp. 156-158, 204, and 206-207, 1984.
DeGloria et al., "A Programmable Instruction Format Extension to VLIW Architectures", Proceedings of CompEuro '92: Computer Systems and Software Engineering, pp. 35-40, May 4-8, 1992.
G.D. Jones and L.D. Larsen, "Selecting Predecoded Instructions with a Surrogate", IBM Technical Disclosure Bulletin, vol. 36, No. 06A, Jun. 1993, pp. 35-37.
G.D. Jones and L.D. Larsen, "Pre-Composed Superscalar Architecture", IBM Technical Bulletin, vol. 37, No. 09, Sep. 1994, pp. 447-451.

ART-UNIT: 273

PRIMARY-EXAMINER: Treat; William M.

ATTY-AGENT-FIRM: Law Offices of Peter H. Priest

ABSTRACT:

A hierarchical instruction set architecture (ISA) provides pluggable instruction set capability and support of array processors. The term pluggable is from the programmer's viewpoint and relates to groups of instructions that can easily be added to a processor architecture for code density and performance enhancements. One specific aspect addressed herein is the unique compacted instruction set which allows the programmer the ability to dynamically create a set of compacted instructions on a task by task basis for the primary purpose of improving control and parallel code density. These compacted instructions are parallelizable in that they are not specifically restricted to control code application but can be executed in the processing elements (PEs) in an array processor. The ManArray family of processors is designed for this dynamic compacted instruction set capability and also supports a scalable array of from one to N PEs. In addition, the ManArray ISA is defined as a hierarchy of ISAs which allows for future growth in instruction capability and supports the packing of multiple instructions within a hierarchy of instructions.

12 Claims, 20 Drawing figures

WEST

Generate Collection Print

L20: Entry 151 of 311 File: USPT Aug 8, 2000

DOCUMENT-IDENTIFIER: US 6101592 A
TITLE: Methods and apparatus for scalable instruction set architecture with dynamic compact instructions

Detailed Description Text (38):
The goal of the packed Load/Store instructions 304 of FIG. 3C is to provide high-density code for moving data between SP registers and memory and PE registers and their local PE memories. In particular, these instructions facilitate rapid context switching for the kernel, and efficient data load/store operations for application tasks. The priorities for selecting load/store addressing modes have been established in the following order:

WEST

Generate Collection | Print

L20: Entry 154 of 311 File: USPT May 30, 2000

US-PAT-NO: 6070237
DOCUMENT-IDENTIFIER: US 6070237 A

TITLE: Method for performing population counts on packed data types

DATE-ISSUED: May 30, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Peleg; Alexander	Haifa			IL
Yaari; Yaakov	Haifa			IL
Mittal; Millind	Haifa			IL
Mennemeier; Larry	Boulder Creek	CA		
Eitan; Benny	Haifa			IL

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Intel Corporation	Santa Clara	CA			02

APPL-NO: 08/ 609899 [PALM]
DATE FILED: March 4, 1996

PARENT-CASE:
CROSS-REFERENCE TO RELATED APPLICATIONS The following co-pending application is related: U.S. patent application entitled: AN APPARATUS FOR PERFORMING A POPULATION COUNT OPERATION, invented by Yaron Ashkenazi, with Ser. No. 08/633,066 filed Apr. 16, 1996, which is a 37 C.F.R. .sctn. 1.60 continuation of application Ser. No. 08/499,095, filed Jul. 6, 1995; which is a 37 C.F.R. .sctn. 1.62 continuation of application Ser. No. 08/175,783, filed Dec. 30, 1995.

INT-CL: [07] G06 F 7/00

US-CL-ISSUED: 712/210; 712/208, 711/210, 711/211
US-CL-CURRENT: 712/210; 711/210, 711/211, 712/208

FIELD-OF-SEARCH: 395/800, 364/DIG.1, 712/210, 712/208, 708/210, 708/211

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
3711692	January 1973	Batcher	364/715.09
3723715	March 1973	Chen et al.	235/175
4161784	July 1979	Cushing et al.	364/748
4189716	February 1980	Krambeck	340/347DD
4393468	July 1983	New	364/736
4418383	November 1983	Doyle et al.	364/200
4498177	February 1985	Larson	371/52
4630192	December 1986	Wassel et al.	364/200
4707800	November 1987	Montrone et al.	364/788
4771379	September 1988	Ando et al.	364/200
4785393	November 1988	Chu et al.	364/200
4785421	November 1988	Takahashi et al.	364/715.04
4901270	February 1990	Galbi et al.	364/786
4989168	January 1991	Kuroda et al.	364/715.09
5095457	March 1992	Jeong	364/758
5187679	February 1993	Vassiliadis et al.	364/786
5201056	April 1993	Daniel et al.	395/800

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
0318957 A3	November 1988	EP	

OTHER PUBLICATIONS

Y. Kawakami et al., LSI Applications: A Single-Chip Digital Signal Processor for Voiceband Applications, Solid State Circuits Conference, Digest of Technical Papers; IEEE International (1980).
J. Shipnes, Graphics Processing with the 88110 RISC Microprocessor, IEEE (1992), pp. 169-174.
Errata to MC88110 Second Generation RISC Microprocessor User's Manual, Motorola Inc. (1992), pp. 1-11.
i860.TM. Microprocessor Family Programmer's Reference Manual, Intel Corporation (1992), Ch. 1, 3, 8, 12.
R. B. Lee, Accelerating Multimedia With Enhanced Microprocessors, IEEE Micro (Apr. 1995), pp 22-32.
Pentium Processor User's Manual, vol. 3: Architecture and Programming Manual, Intel Corporation (1993), Ch. 1, 3, 4, 6, 8 and 18.
N. Margulis, i860 Microprocessor Architecture, McGraw Hill, Inc. (1990) Ch. 6, 7, 8, 10, 11.
Titled "MC88110 Second Generation RISC Microprocessor User's Manual" Sep. 1992, pp. 1-23, 2-1 to 2-20, 3-1 to 3-32, 5-1 to 5-25, 10-62 to 10-71, Index 1 to 17.
Motorola Semiconductor Technical Data Titled "MC88110 Programmer's Reference Guide", Dec. 1992, pp. 1-4.
Titled "Intel i 750.RTM., i860.TM., i960.RTM. Processors and Related Products", 1993, pp. 1-3.
Sun Microsystems, Inc., SPARC Technology Business Ultra SPARC, UltraSPARC Multimedia Capabilities On-Chip Support for Real-Time Video and Advanced Graphics, Sep. 1994 8 pages.
Microprocessor Report, Brian Case, Philips Hopes to Displace DSPs with VLIW: TriMedia Processors Aimed at Future Multimedia Embedded Apps, Dec. 5, 1994, pp. 12-18.
Microprocessor Report, Linley Gwennap, New PA-RISC Processor Decodes MPEG Video: HP's PA-7100LC Uses New Instructions to Eliminate Decoder Chip, Jan. 24, 1994, pp. 16-17.

IBM Technical Disclosure Bulletin, vol. 35 No. 1A, Bit Zone Accumulator, Jun. 1992, p. 106.
Mc88110 Second Generation RISC Microprocessor User's Manual, Motorola, Inc. (1991).
TMS320C2x User's Guide, Texas Instruments (1993) pp 3-2 through 3-11; 3-28 through 3-34; 4-1 through 4-22; 4-41;4-103;4-199 through 4-120; 4-122; 4-150 through 4-151.

ART-UNIT: 273

PRIMARY-EXAMINER: An; Meng-Ai T.

ASSISTANT-EXAMINER: Nguyen; Dzung C.

ATTY-AGENT-FIRM: Blakely, Sokoloff, Taylor & Zafman LLP

ABSTRACT:

A novel processor for manipulating packed data. The packed data includes a first data element D1 and a second data element D2. Each of said data elements has a predetermined number of bits. The processor comprises a decoder, a register, and a circuit. The decoder is for decoding a control signal responsive to receiving the control signal. The register is coupled to the decoder. The register is for storing the packed data. The circuit is coupled to the decoder. The circuit is for generating a first result data element R1 and a second data element R2. The circuit is further for generating R1 to represent a total number bits set in D1, and the circuit is further for generating R2 to represent a total number bits set in D2.

13 Claims, 16 Drawing figures

WEST

Generate Collection | Print

L20: Entry 154 of 311 File: USPT May 30, 2000

DOCUMENT-IDENTIFIER: US 6070237 A
TITLE: Method for performing population counts on packed data types

CLAIMS:

1. A computer-implemented method comprising:

a) decoding an instruction, the instruction indicating a storage location of a first packed data sequence having a set of packed data elements, said instruction operable to specify a variable quantity of said packed data elements, said instruction operable to specify a variable size of said packed data elements, and said instruction specifying an operation to be performed on said packed data elements;

b) generating, in response to executing said instruction, a result packed data sequence having a set of result packed data elements corresponding to said set of packed data elements of said first packed data sequence, said result packed data elements respectively representing population counts of a number of bits set in said packed data elements of said first packed data sequence.

WEST

Generate Collection | Print

L20: Entry 172 of 311 File: USPT Dec 28, 1999

US-PAT-NO: 6009263
DOCUMENT-IDENTIFIER: US 6009263 A

TITLE: Emulating agent and method for reformatting computer instructions into a standard uniform format

DATE-ISSUED: December 28, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Golliver; Roger A.	Beaverton	OR		
Doshi; Gautam Bhagwandas	Sunnyvale	CA		
Huck; Jerome C.	Palo Alto	CA		
Karp; Alan Hersh	Palo Alto	CA		
Makineni; Sivakumar	Sunnyvale	CA		
Morrison; Mike	Santa Clara	CA		
Colon-Bonet; Glen	Fort Collins	CO		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Institute For The Development Of Emerging Architectures, L.L.C.	Cupertino	CA			02

APPL-NO: 08/ 901471 [PALM]
DATE FILED: July 28, 1997

PARENT-CASE:
RELATED APPLICATIONS The present application is related to co-pending application entitled "Method, Apparatus and Computer System for Directly Transferring and Translating Data Between an Integer Processing Unit and a Floating Point Processing Unit," filed on Oct. 10, 1996, Ser. No. 08/728,646.

INT-CL: [06] G06 F 9/30, G06 F 9/45

US-CL-ISSUED: 395/500.48; 395/500.47, 395/707, 708/204, 709/202, 712/200
US-CL-CURRENT: 703/27; 703/26, 708/204, 709/202, 712/200, 717/138

FIELD-OF-SEARCH: 395/500, 395/376, 395/379, 395/705, 395/800.01, 395/707, 395/500.44, 395/500.48, 395/500.49, 395/500.47, 364/715.03, 364/748.01, 364/748.19, 364/748.16, 712/200, 712/210, 708/204, 708/495, 709/202

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	4833599	May 1989	Coldwell et al.	364/200
☐	5268855	December 1993	Mason et al.	364/748
☐	5574927	November 1996	Scantlin	395/800
☐	5699536	December 1997	Hopkins et al.	395/392
☐	5764959	June 1998	Sharangpani et al.	395/500
☐	5805475	September 1998	Putrino et al.	364/715.03
☐	5930495	July 1999	Christopher et al.	395/500

ART-UNIT: 273

PRIMARY-EXAMINER: Teska; Kevin J.

ASSISTANT-EXAMINER: Phan; Thai

ATTY-AGENT-FIRM: Blakely Sokoloff Taylor & Zafman, LLP

ABSTRACT:

An emulating agent and method is provided that receives numbers having si, exponents and significands of varying lengths and possibly configured in a variety of incompatible formats and to reformat the numbers into a standard uniform format for uniform arithmetic computations in processors operating with different architectures. In one embodiment, the emulating agent has a three-field superset register configured to receive the sign of a number in a first field, the exponent of a number in a second field and the significand of a number in a third field, regardless of the original format of the number, resulting in a number represented in a standard uniform format for computation. The embodiment also allows high level access to the fields to allow users to control the size of the numbers inserted into the fields.

14 Claims, 4 Drawing figures

WEST

Generate Collection | Print

L20: Entry 172 of 311 File: USPT Dec 28, 1999

DOCUMENT-IDENTIFIER: US 6009263 A
TITLE: Emulating agent and method for reformatting computer instructions into a standard uniform format

Detailed Description Text (21):
Memory access instructions are required in order for proper format conversion. Table 4 illustrates a sample of instructions for memory access in different operations involved in the format conversion. There are separate floating-point load and store instructions for the single, double and double extended floating-point real data type and the packed signed or unsigned integer data. In a preferred embodiment, the addressing modes and memory hint options for floating-point load and store instructions are the same with the integer load and store instructions. Table 4 illustrates a list of sample floating-point load/store instructions.

WEST

Generate Collection | Print

L20: Entry 198 of 311 File: USPT Jun 1, 1999

US-PAT-NO: 5909572
DOCUMENT-IDENTIFIER: US 5909572 A

TITLE: System and method for conditionally moving an operand from a source register to a destination register

DATE-ISSUED: June 1, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Thayer; John S.	Houston	TX		
Favor; John G.	Scotts Valley	CA		
Weber; Frederick D.	San Jose	CA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Compaq Computer Corp.	Houston	TX			02
Advanced Micro Device, Inc.	Sunnyvale	CA			02

APPL-NO: 08/ 759025 [PALM]
DATE FILED: December 2, 1996

INT-CL: [06] G06 F 9/00

US-CL-ISSUED: 395/567; 395/564, 395/566
US-CL-CURRENT: 712/226; 712/223, 712/225

FIELD-OF-SEARCH: 395/567, 395/564, 395/566, 395/562, 395/568, 395/563, 395/595, 395/800.35

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	3701977	October 1972	Mendelson et al.	711/126
☐	4707800	November 1987	Montrone et al.	364/788
☐	4725973	February 1988	Matsuura et al.	364/736.03
☐	4782441	November 1988	Inagami et al.	395/800.09
☐	4849882	July 1989	Aoyama et al.	395/800.06
☐	4884197	November 1989	Sachs et al.	711/123
☐	4891754	January 1990	Boreland	395/586
☐	5025407	June 1991	Gulley et al.	364/748.2
☐	5073864	December 1991	Methvin et al.	364/715.11
☐	5181183	January 1993	Miyazaki	364/725.03
☐	5187796	February 1993	Wang et al.	395/800.04
☐	5193167	March 1993	Sites et al.	711/163
☐	5226171	July 1993	Hall et al.	395/800.09
☐	5235536	August 1993	Matsubishi et al.	364/736.04
☐	5251323	October 1993	Isobe	395/800.05
☐	5307300	April 1994	Komoto et al.	364/736.01
☐	5335330	August 1994	Inoue	395/588
☐	5434592	July 1995	Dinwiddie, Jr. et al.	345/133
☐	5437043	July 1995	Fujii et al.	395/800.1
☐	5453945	September 1995	Tucker	364/725.01
☐	5471637	November 1995	Pawlowski et al.	394/296
☐	5511219	April 1996	Shimony et al.	395/800.35
☐	5513366	April 1996	Agrawal et al.	395/800.22
☐	5537640	July 1996	Pawlowski et al.	711/146
☐	5627981	May 1997	Adler et al.	395/582
☐	5640588	June 1997	Vegesna et al.	395/800.23
☐	5644520	July 1997	Pan et al.	364/736.01
☐	5655096	August 1997	Branigin	395/376
☐	5669013	September 1997	Watanabe et al.	395/825
☐	5673408	September 1997	Shebanow et al.	395/392
☐	5673426	September 1997	Shen et al.	395/591
☐	5692211	November 1997	Gulick et al.	395/800.35
☐	5745721	April 1998	Beard et al.	395/384
☐	5801975	September 1998	Thayer et al.	364/725
☐	5850227	December 1998	Longhenry et al.	345/439

OTHER PUBLICATIONS

A comparison of full and partial predicated execution support for ILP processors by Mahlke et al., 1995 IEEE publication, pp. 138-149.
Kohn, L, et al., "The Visual Instruction Set (VIS) in UltraSPARC," SPARC Technology Business--Sun Microsystems, Inc., 1996 IEEE pp. 462-489.
Gwennap, Linley, "UlatraSparc Adds Multimedia Instructions--Other New Instructions Handle Unaligned and Little-Endian Data," Microprocessor Report, Dec. 5, 1994, pp. 16-18.

Lee, Ruby B., "Realtime PMEG Video via Software Decompression on a PA-RISC Processor," Hewlett-Packard Company, 1995 IEEE, pp. 186-192.
Mattison, Phillip E., "Practical Digital Video With Programming Examples in C, " Wiley Professional Computing, pp. 158-178.
Zhou, Chang-Guo, et al., "MPEG Video Decoding With the UltraSPARC Visual Instruction Set," Sun Microsystems Inc., 1995 IEEE, pp. 470-474.

ART-UNIT: 278

PRIMARY-EXAMINER: Maung; Zarni

ATTY-AGENT-FIRM: Conley, Rose & Tayon Kowert; Robert C. Daffer; Kevin L.

ABSTRACT:

A multimedia extension unit (MEU) is provided for performing various multimedia-type operations. The MEU can be coupled either through a coprocessor bus or a local CPU bus to a conventional processor. The MEU employs vector registers, a vector ALU, and an operand routing unit (ORU) to perform a maximum number of the multimedia operations within as few instruction cycles as possible. Complex algorithms are readily performed by arranging operands upon the vector ALU in accordance with the desired algorithm flowgraph. The ORU aligns the operands within partitioned slots or sub-slots of the vector registers using vector instructions unique to the MEU. At the output of the ORU, operand pairs from vector source or destination registers can be easily routed and combined at the vector ALU. The vector instructions employ special load/store instructions in combination with numerous operational instructions to carry out concurrent multimedia operations on the aligned operands.

12 Claims, 19 Drawing figures

WEST

Generate Collection | Print

L20: Entry 198 of 311 File: USPT Jun 1, 1999

DOCUMENT-IDENTIFIER: US 5909572 A
TITLE: System and method for conditionally moving an operand from a source register to a destination register

Brief Summary Text (41):
Arithmetic scaling which is lacking from many conventional operations is readily performed as part of the present load/store instructions. For example, packing and unpacking instructions found in many DSP instruction sets can be avoided. Thus, unpacking of an 8-bit word into a 20-bit slot occurs as part of a load instruction, whereas packing of a 20-bit operand to an 8-bit word occurs as part of the store instruction. Combining packing and unpacking operations into store and load helps eliminate unnecessary move operations which occur as part of stand-alone conventional pack and unpack instructions.

Detailed Description Text (116):
The interleave mapping for 10-bit partitions is completely transparent to the programmer as long as only 10-bit loads/stores and vector instructions are performed on a given set of data. Interleaved mapping of 20-bit partitions is also transparent to the programmer if only 20-bit operations are performed. However, if 10-bit and 20-bit operations are mixed, then care must be taken to understand the mapping so that the expected results are produced. The interleaving can be very useful, for example, if a 10-bit load from an octet-sized memory location automatically expands and interleaves the byte-wide memory data to the upper portion of 20-bit partitions. The 20-bit operation can be immediately performed on this data without the need for explicit format conversions. Subsequently, 10-bit stores to octets can automatically perform the inverse 20-bit to 10-bit packing function. Thus, the present store operation, namely vstb mem64, vsh performs packing of n+4 bits within a slot of a vector register to n/2 bits within an address of the memory unit. Given n=16, 20-bit-to-8-bit packing can occur as part of the store operation. Additional operations, such as move or shift operations need not occur to perform a packing function. Packing serves to store the most significant bits from a slot. Unpacking is an operation by which n/2 bits from a memory address are loaded into n+4 bit locations within a slot. If n=16, then a load operation such as vldb vdh, mem64 causes 8-bits within a memory address to be loaded into a 20-bit slot. Utilizing load and store functions in such a manner thereby avoids having to implement separate unpack and pack instructions, respectively, within the MEU instruction set. Accordingly, the same result can be achieved but with fewer instructions. For MPEG, 8-bit pixels are unpacked to 20-bit numbers for DCT or IDCT manipulations, then the results are repacked to 8-bit pixels. The internals of the DCT and IDCT operations require more than 8 bits of precision, to which packing and unpacking are particularly advantageous.

WEST

Generate Collection | Print

L20: Entry 289 of 311 File: USPT Aug 23, 1983

US-PAT-NO: 4400776
DOCUMENT-IDENTIFIER: US 4400776 A

TITLE: Data processor control subsystem

DATE-ISSUED: August 23, 1983

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bazlen; Dieter	Stuttgart			DE
Bock; Dietrich W.	Schoenaich			DE
Getzlaff; Klaus J.	Boeblingen			DE
Hajdu; Johann	Boeblingen			DE
Painke; Helmut	Boeblingen			DE

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
International Business Machines Corporation	Armonk	NY			02

APPL-NO: 06/ 186883 [PALM]
DATE FILED: September 12, 1980

FOREIGN-APPL-PRIORITY-DATA:

COUNTRY	APPL-NO	APPL-DATE
DE	2936801	September 12, 1979

INT-CL: [03] G06F 9/00

US-CL-ISSUED: 364/200
US-CL-CURRENT: 712/208

FIELD-OF-SEARCH: 364/200, 364/900

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
☐	3426328	February 1969	Gunderson et al.	364/200
☐	3611307	October 1971	Podvin	364/200
☐	3614745	October 1971	Podvin	364/200
☐	3962683	June 1976	Brown et al.	364/200
☐	4040021	August 1977	Birchall et al.	364/200
☐	4047245	September 1977	Knipper	364/200

OTHER PUBLICATIONS

Albers, Variable Radix Shift Counter, IBM Technical Disclosure Bulletin, vol. 19, No. 11, Apr. 1977, pp. 4201-4202.
Barton et al, Clock Pulse Stretching Technique, IBM Technical Disclosure Bulletin, vol. 21, No. 8, Jan. 1979, pp. 3218-3219.
Spengler, Clock Circuit, IBM Technical Disclosure Bulletin, vol. 18, No. 3, Aug. 1975, pp. 867-868.

ART-UNIT: 236

PRIMARY-EXAMINER: Nusbaum; Mark E.

ASSISTANT-EXAMINER: Fleming; Michael R.

ATTY-AGENT-FIRM: Conley; Gregory A. Galbi; E. W. Jancin, Jr.; J.

ABSTRACT:

An improved data processor control subsystem in which a cycle counter having a plurality of cascade-connected stages also comprises one or more supplemental or dummy stages, which can be selectively inserted or removed from the chain of cascade-connected stages, to alter the number of sub-cycles in an operating cycle, thereby decreasing the complexity of associated decoding circuitry.

5 Claims, 9 Drawing figures

WEST

Generate Collection | Print

L20: Entry 289 of 311 File: USPT Aug 23, 1983

DOCUMENT-IDENTIFIER: US 4400776 A
TITLE: Data processor control subsystem

Detailed Description Text (20):
The above given example of an instruction by means of which data are to be transferred from main store 1 to local store 28 either in unpacked or in packed form shows that for otherwise identical processes both types of microinstructions differ only in that one cycle time in which the data are transformed from an unpacked into a packed form.

Detailed Description Text (29):
The specific feature of instruction cycle counter 22a consists in that this counter which is e.g. designed for the simple control of a microinstruction which fetches data from the main store and transfers them to the local store, the conversion of the data from an unpacked into a packed form being possibly included in the control, also comprises an additional flipflop 54 activated upon request only, said flipflop generating the additional cycle time TZ. Output lines 59 of the respective stages are connected to the various gates of the data flow where the various cycle times, combined with the output signals of operation decoder 15 perform the control actions in the execution of the respective microinstruction. The combination of the control signals, i.e. of the output signals of operation decoder 15 with the respective cycle times is not shown in detail in FIG. 1 but can be concluded from FIG. 3.

Detailed Description Text (32):
Microinstructions whose data cover a longer path from source to origin, as e.g. in an instruction which could be: "Fetch decimal data, convert them into the packed form and transfer them to local store" generate a control signal on line 57, so that now with an enabled gate 52 the additional flipflop 54 can be inserted via OR gate 55 into the flipflop chain between the flip-flop for cycle time T4 and the flipflop for cycle time T5. The direct path of the activation signal is blocked via inverter 51 and the not enabled gate 53. In this manner, the additional cycle time required for converting the decimal data into a packed form is generated for the data propagation on the longer path.

WEST

Generate Collection | Print

L20: Entry 298 of 311 File: USPT Jul 25, 1978

US-PAT-NO: 4103329
DOCUMENT-IDENTIFIER: US 4103329 A

TITLE: Data processing system with improved bit field handling

DATE-ISSUED: July 25, 1978

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Davis; Michael Ian	Kings Worthy near Winchester			GB
Hood; Robert Allen	Boca Raton	FL		
Mayes; Gary Wayne	Boca Raton	FL		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
International Business Machines Corporation	Armonk	NY			02

APPL-NO: 05/ 755105 [PALM]
DATE FILED: December 28, 1976

INT-CL: [02] G06F 9/06

US-CL-ISSUED: 364/200
US-CL-CURRENT: 712/300

FIELD-OF-SEARCH: 364/2MSFile, 364/9MSFile

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

Search Selected | Search ALL

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
3401375	September 1968	Bell et al.	364/200
3614746	October 1971	Klinkhamer	364/200
3624616	November 1971	Patel	364/200
3654621	April 1972	Bock et al.	364/200

ART-UNIT: 237

PRIMARY-EXAMINER: Springborn; Harvey E.

ATTY-AGENT-FIRM: Gershuny; Edward S.

ABSTRACT:

Hardware facilities are described whereby the handling of data represented by variable length fields of bits may be made faster, use less storage and be less prone to errors in programming. The bit fields are handled independently of the natural storage addressing elements and boundaries. Data may be packed into main storage with the highest efficiency, and manipulated with a fast and efficient hardware instruction set.

4 Claims, 28 Drawing figures

WEST

Generate Collection Print

L20: Entry 298 of 311 File: USPT Jul 25, 1978

DOCUMENT-IDENTIFIER: US 4103329 A
TITLE: Data processing system with improved bit field handling

Abstract Text (1):
Hardware facilities are described whereby the handling of data represented by variable length fields of bits may be made faster, use less storage and be less prone to errors in programming. The bit fields are handled independently of the natural storage addressing elements and boundaries. Data may be packed into main storage with the highest efficiency, and manipulated with a fast and efficient hardware instruction set.

Drawing Description Text (17):
FIG. 17 shows an example of the use of the "load field and increment" instruction to access variable length bit fields within packed data; and

WEST Search History

DATE: Monday, November 17, 2003

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,PGPB; PLUR=NO; OP=ADJ			
L20	l14 not L19	311	L20
L19	l14 not l17	2118	L19
L18	l14 not l17L17	2429	L18
L17	l15 not package$1	246928	L17
L16	l14 and L15	753	L16
L15	l12 not (packet$1)	453221	L15
L14	instruction$1 with L13	2429	L14
L13	(load$3 or stor$3) with L12	72868	L13
L12	pack$3 or unpack$3 or un-pack$3	533925	L12
L11	load lower byte	4	L11
L10	load packed byte	0	L10
L9	load unpacked data	0	L9
DB=JPAB,EPAB,DWPI,TDBD; PLUR=NO; OP=ADJ			
L8	load packed word	0	L8
L7	load packed byte	0	L7
L6	load lower byte	0	L6
L5	load upper byte	0	L5
L4	load packed data	0	L4
L3	load unpacked data	0	L3
L2	unpack instruction$1	8	L2
DB=USPT,PGPB; PLUR=NO; OP=ADJ			
L1	unpack instruction$1	67	L1

END OF SEARCH HISTORY